

# Fast neural-net based fake track rejection in the LHCb reconstruction

Michel De Cian<sup>1</sup>, Stephen Farry<sup>2</sup>, Paul Seyfert<sup>3,\*</sup>, Sascha Stahl<sup>4</sup>.

<sup>1</sup>*Physikalisches Institut, Ruprecht-Karls-Universität Heidelberg, Heidelberg, Germany*

<sup>2</sup>*Oliver Lodge Laboratory, University of Liverpool, Liverpool, United Kingdom*

<sup>3</sup>*Sezione INFN di Milano Bicocca, Milano, Italy*

<sup>4</sup>*European Organization for Nuclear Research (CERN), Geneva, Switzerland*

*\* now at <sup>4</sup>*

## Abstract

A neural-network based algorithm to identify fake tracks in the LHCb pattern recognition is presented. This algorithm, called ghost probability, retains more than 99% of well reconstructed tracks while reducing the number of fake tracks by 60%. It is fast enough to fit into the CPU time budget of the software trigger farm and thus reduces the combinatorics of the decay reconstructions, as well as the number of tracks that need to be processed by the particle identification algorithms. As a result, it strongly contributes to the achievement of having the same reconstruction online and offline in the LHCb experiment in Run II of the LHC.



# 1 Introduction

The LHCb detector consists of subsystems designed to perform high efficiency track reconstruction ( $> 95\%$ ) [1] with an excellent momentum resolution ( $0.5\%$  for  $p < 20 \text{ GeV}/c$ ) [2]. Two Ring Imaging Cherenkov detectors provide precise particle identification. In Run II of the LHC (2015–2018), a new scheme for the LHCb software trigger allows splitting the triggering of the event into two stages, giving room to perform the alignment and calibration in real time. In the novel detector alignment and calibration strategy for Run II, data collected at the start of the fill are processed in a few minutes and used to update the alignment, while the calibration constants are evaluated for each run<sup>1</sup> [3]. This allows identical alignment and calibration constants to be used in the online and offline reconstruction.

A major achievement of LHCb’s operations in Run II is that the full offline reconstruction is run in the second stage of the software trigger. In combination with the final offline calibration available in the software trigger, this results in the best possible reconstruction quality for online event selection.

One of the challenges to run the full offline reconstruction in the software trigger is the limited CPU time budget of the computing farm. The reconstruction time of events depends strongly on the number of reconstructed charged particle tracks in an event in two ways. Firstly, the particle identification (PID) is evaluated for every reconstructed long track<sup>2</sup> in the second stage of the software trigger. Secondly, there are more possible track combinations to consider in the reconstruction of decay vertices with more reconstructed tracks.

A key ingredient to fit the offline reconstruction into the software trigger is the reduction of the fake track rate prior to the PID and decay reconstruction in the second software trigger stage. Fake tracks are defined as those reconstructed tracks which do not correspond to the trajectory of a true particle but instead are due to the mismatch of hits from separate particles or from detector noise. A neural network, described in this note, is deployed to identify these fake tracks, called the “ghost probability”<sup>3</sup>.

## 2 Terminology

To avoid ambiguity, the bare term “performance” is avoided. Instead, when referring to how well good tracks are separated from fake tracks, the term “physics performance” is used since it is the figure of merit on which physics analyses depend. The term “CPU performance” is used for the amount of computing resources needed to execute the algorithm described in this note. As benchmark for the latter, the cycle count of callgrind [4] is used. Effects of instruction caching and data caching are assumed to be small, approximately confirmed by wall clock time measurements. The cycles spent in

---

<sup>1</sup>A run in LHCb is an up-to one hour long data taking interval – the smallest subdivision of an LHC fill in LHCb.

<sup>2</sup>Full PID for downstream tracks was gradually introduced in the software trigger.

<sup>3</sup>The name is due to the fact that fake tracks are commonly called “ghosts” in LHCb.

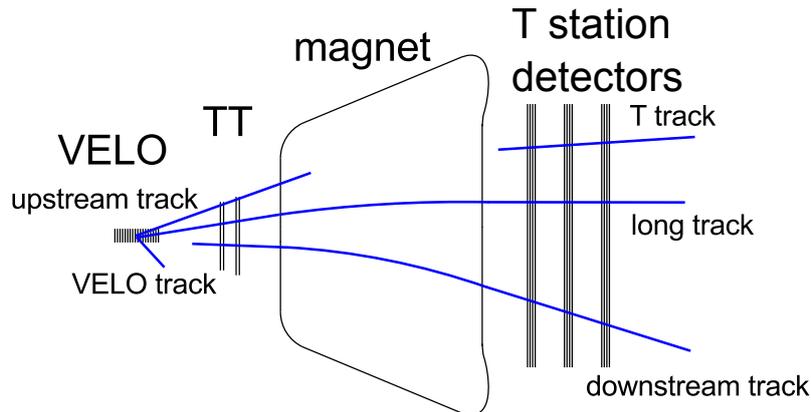


Figure 1: Illustration of the tracking system of LHCb, starting from the VELO around the collision point on the left, particles pass the TT, are deflected in the magnetic field of the dipole magnet and then detected in the T station detector (IT and OT). Different track types are reconstructed by different track finding algorithms. [1]

37 other algorithms which are *only* called to compute input quantities to the ghost probability  
 38 are assigned to the ghost probability, most notably this includes the interpolation of tracks  
 39 through active detector material to determine which channels should have a hit from  
 40 the track – other algorithms which compute input quantities, which would be executed  
 41 anyways, like the track fit, are not assigned to the ghost probability.

42 The term “ghost probability” is used for both the entire algorithm computing whether  
 43 a track is considered a fake track or a real track, including the neural network, and for  
 44 the numeric response of that algorithm. When the ghost probability is referred to as a  
 45 selection requirement, the nominal working point corresponding to a fake track retention  
 46 of 40 % is implied.

## 47 1.1 Tracking system and track reconstruction

48 The LHCb detector is a single-arm forward spectrometer covering the pseudorapidity  
 49 range  $2 < \eta < 5$ , designed for the study of particles containing  $b$  or  $c$  quarks. The detector  
 50 includes a high-precision tracking system consisting of a silicon-strip vertex detector  
 51 (VELO) surrounding the  $pp$  interaction region, a large-area silicon-strip detector (TT)  
 52 located upstream of a dipole magnet with a bending power of about 4 Tm, and three  
 53 stations (T stations) of silicon-strip detectors (close to the beam pipe, called IT) and  
 54 straw drift tubes (further away from the beam pipe, called OT) placed downstream of the  
 55 magnet.

56 The readout of the silicon-strip detectors provides a measurement of the amplitude  
 57 of the signal induced in the detector [5]. The readout of the OT provides a drift-time  
 58 measurement with respect to the beam crossing signal which translates to a measurement  
 59 of the distance from the anode wire at which a charged particle passed through a straw

60 tube [6].

61 Different types of charged hadrons are distinguished using information from two  
62 ring-imaging Cherenkov detectors. Photons, electrons and hadrons are identified by a  
63 calorimeter system consisting of scintillating-pad and preshower detectors, an electromag-  
64 netic calorimeter and a hadronic calorimeter. Muons are identified by a system composed of  
65 alternating layers of iron and multiwire proportional chambers. The online event selection  
66 is performed by a trigger, which consists of a hardware stage (L0), based on information  
67 from the calorimeter and muon systems, followed by the two-staged software stage (HLT1  
68 and HLT2), which applies a full event reconstruction.

69 Owing to the design of the LHCb detector, which consists of tracking detectors mainly  
70 outside the magnetic field, charged particle tracks are approximately straight line segments  
71 in the upstream part (VELO and TT) and in the downstream part (T stations). Figure 1  
72 shows an overview of the different track types defined in the LHCb reconstruction: VELO  
73 tracks, which have hits in the VELO only; upstream tracks, which have hits in the two  
74 upstream trackers; T tracks, which have hits in the T stations only; downstream tracks,  
75 which have hits in TT and the T stations; and long tracks, which have hits in the VELO  
76 and the T stations. The latter tracks can additionally have hits in TT.

77 If a particle is reconstructed more than once, as different track types, only the track  
78 best suited for analysis purposes is kept. Hereby, long tracks are preferred over any other  
79 track type, upstream tracks are preferred over VELO tracks, and downstream tracks are  
80 preferred over T tracks.

81 Most analyses use long tracks because they provide the best momentum and spatial  
82 resolution among all track types. Unless otherwise stated, track reconstruction at LHCb  
83 refers to the reconstruction of long tracks. In a typical signal triggered event, around 60  
84 long tracks are reconstructed. Other track types, such as downstream tracks, are used  
85 for the reconstruction of decay products of long-lived particles such as  $K_s^0$  mesons, or for  
86 internal alignment of the tracking detectors.

87 Tracks are fit with a Kalman filter. In addition to a global fit  $\chi^2$ , separate contributions  
88 to the  $\chi^2$  from the downstream detectors (IT and OT),  $\chi_D^2$ , and from the upstream  
89 detectors (VELO and TT),  $\chi_U^2$  are computed. A large number of fake tracks result from  
90 wrong combinations of well reconstructed track segments in the upstream and downstream  
91 regions. These usually have good  $\chi_D^2$  and  $\chi_U^2$  but the additional contribution from matching  
92 the two segments,  $\chi_M^2 = \chi^2 - \chi_D^2 - \chi_U^2$ , is large for these “matching” fakes.

93 The Kalman fit performs an outlier removal to account for individual detector hits  
94 which are not due to the reconstructed particle. Beyond that, a special treatment for OT  
95 hits is in place. The readout electronics is designed to select only a single hit in each  
96 channel per bunch crossing; if two charged particles pass the same straw, a drift-time  
97 measurement will only be provided for one of them. To describe tracks in high occupancy  
98 OT modules, the drift-time measurement can be ignored and only the information that  
99 a track went somewhere through the straw is used. This is decided for each straw-track  
100 combination individually if the hit residual is too large, similar to a standard outlier  
101 removal. This drift-time suppression ensures that the track fit  $\chi^2$  is not biased to larger

102 values for tracks in high multiplicity events, for tracks in the OT with respect to tracks in  
 103 the IT, or for tracks in high occupancy modules, which are those closer to the beam axis.

104 Fake long tracks can occur at different places in the pattern recognition. Most of  
 105 the fake long tracks originate from particles which cannot be tracked through the entire  
 106 detector due to a hadronic interaction with the detector material, followed by fake track  
 107 segments in the T stations and tracks where two different particles were reconstructed  
 108 in the upstream and downstream parts of the detector. Fake reconstruction of VELO  
 109 segments occurs at a lower rate. Mismatched track segments are due to the long lever arm  
 110 between the tracking stations up- and downstream of the magnet and remain to be the  
 111 most abundant category after fake track rejection.

## 112 1.2 Previous works

113 An earlier version of the work presented here [7], referred to as the “old ghost probability”,  
 114 was already used in analyses of Run I (2010–2012) data. It was evaluated in the offline  
 115 reconstruction to distinguish fake tracks from real particles’ *e.g.* in Ref. [8]. The network  
 116 was trained on all reconstructed tracks in simulated events with at least one  $b\bar{b}$  pair  
 117 produced in the  $pp$  collision.

118 The 22 input variables to the old ghost probability are the overall track fit  $\chi^2$  along  
 119 with the individual contributions  $\chi_D^2, \chi_U^2, \chi_M^2$  and the corresponding degrees of freedom for  
 120 each fit; the numbers of hits on the track in each tracking detector; the reconstructed track  
 121  $p_T$  and pseudorapidity; the difference in the number of observed hits on a track and the  
 122 “expected hits”, calculated by interpolating the track through the detector and counting  
 123 how many active strips/straws the track passes through; and finally the occupancies of all  
 124 tracking detectors.

125 There are separate networks for each track type, where input variables are removed if  
 126 they are not defined for that track type (*e.g.* VELO hits for downstream tracks).

## 127 1.3 Network architecture tuning

128 As framework for the neural network, the TMVA package [9] is chosen since it is

- 129 • equipped with a root file interface for the training, which is the common data file  
 130 format in LHCb software,
- 131 • commonly known in LHCb (ensuring future maintainability),
- 132 • able to provide code generation for the trained network such that the network can  
 133 be integrated into any C++ code without creating dependency on external libraries.

134 Mathematically, the shallow neural network is implemented as composed function

$$\mathbb{R}^n \xrightarrow{\text{linear} + \text{const}} \mathbb{R}^m \xrightarrow{\text{element wise non-linear}} \mathbb{R}^m \xrightarrow{\text{linear} + \text{const}} \mathbb{R} \xrightarrow{\text{non-linear}} \mathbb{R}.$$

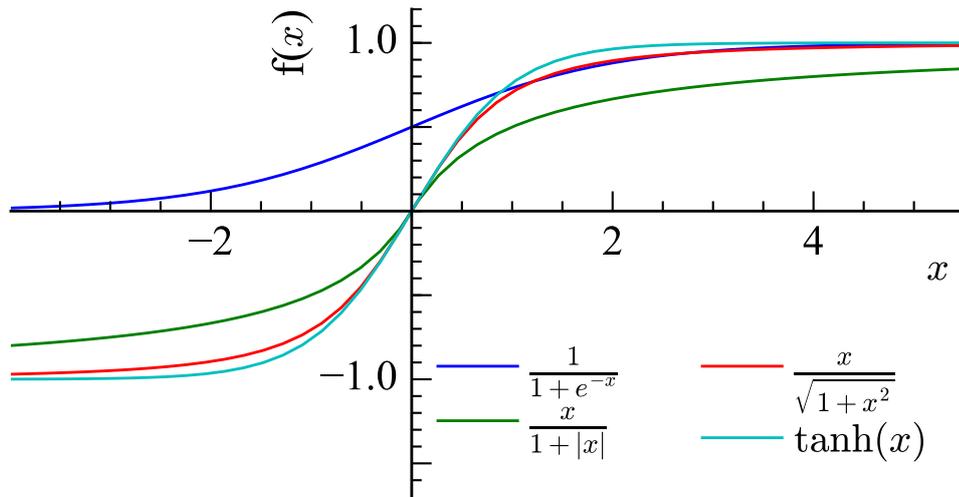


Figure 2: Functional shape of sigmoid functions.

135 With operations in each step:

$$\begin{aligned}
 (x_i) &\mapsto \left( \sum_i M_{j,i} x_i + b_j \right) \\
 (x'_j) &\mapsto (f(x'_j)) \\
 (x''_j) &\mapsto \sum_j M'_{1,j} x''_j + b' \\
 x''' &\mapsto f'(x''')
 \end{aligned}$$

136 The parameters of the linear mappings  $(M_{j,i}, b_j, M'_{1,j}, b')$  are subject to optimisation,  
 137 where  $1 \leq i \leq n$ ,  $1 \leq j \leq m$ , and  $n$  is the number of input variables and  $m$  is chosen as  
 138  $n + 5$ . The non-linearities ( $f$  and  $f'$ ), so-called “activation functions”, are fixed real valued  
 139 functions.

140 At the time of development, the  $\tanh(x)$  function was a commonly used activation  
 141 function in TMVA, while known as a computationally expensive function to be optimised  
 142 for the LHCb pattern recognition [10]. Yet it is not the only possible sigmoid function [11]  
 143 and consequently custom activation functions have been added to TMVA [12].

144 Of the tested functions  $\frac{x}{\sqrt{1+x^2}}$  is the fastest to compute, while no significant physics  
 145 performance difference is expected given the similar functional shape, see Fig. 2. In Fig. 3  
 146 even a small physics performance improvement is visible, possibly due to its larger gradient  
 147 with respect to  $\tanh(x)$  [13]. Therefore, it is chosen as activation function.

148 For additional CPU performance increase, the TMVA generated code is optimised by  
 149 hand to allow auto-vectorisation, move parts of the input variable transformations from  
 150 run time to compile time, and reduce avoidable memory allocations and data copying [14].  
 151 The evaluation of the neural network itself is thereby sped up by a factor two.

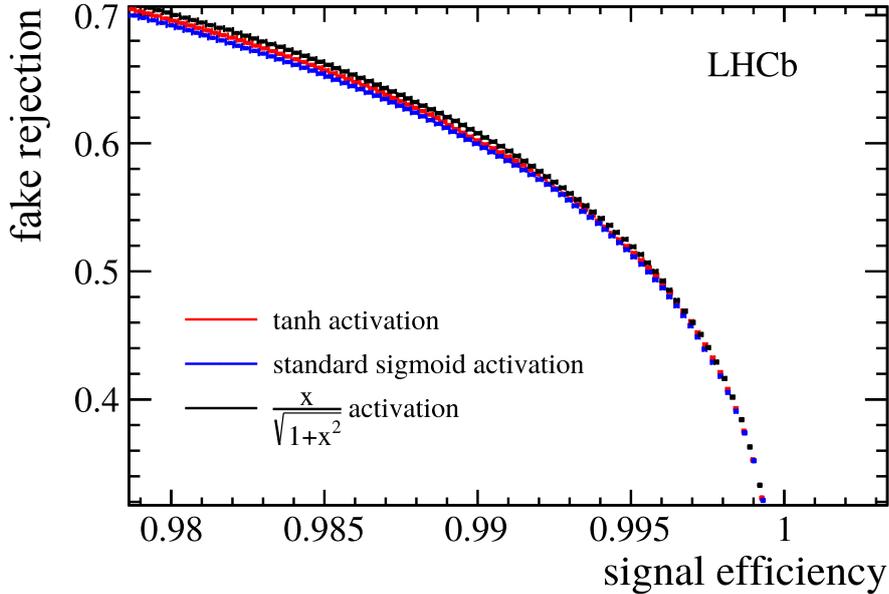


Figure 3: ROC curves for fake track discriminating neural networks, using different activation functions. A very small physics performance improvement is observed when changing from the TMVA standard functions to  $\frac{x}{\sqrt{1+x^2}}$ .

152 The ghost probability is a classification problem, and thus cross entropy [15] is chosen  
 153 as the loss function in the network training. With respect to the Run I implementation,  
 154 where a mean-square-error loss was chosen, this contributes to the physics performance  
 155 improvement. The activation function of the output layer is  $\frac{1}{1-e^{-x}}$  in the training. In the  
 156 application, a custom output calibration is applied instead, as described in Sect. 1.5.

#### 157 1.4 Variable selection

158 To allow for enough development time for testing and evaluation, the selection of input  
 159 variables is mostly unchanged from Run I with two exceptions. The track interpolation to  
 160 determine the number of expected hits is removed to reduce the CPU usage of the ghost  
 161 probability by a factor 10. The number of track candidates competing for shared hits in  
 162 the pattern recognition is added as input variable.

#### 163 1.5 Output transformation

164 To ease the usage of the ghost probability, a transformation of the network response  
 165 is applied. A probability integral transform – commonly referred to as “flattening” or  
 166 “rarity transformation” – is implemented as a linear spline fit to the cumulative network  
 167 response for fake tracks in simulated events. The discriminating behaviour of any classifier  
 168 is invariant under monotonous transformations and so is the physics performance under

169 the probability integral transform. Motivations for this transformation are primarily to  
170 give a physical interpretation to the response: rejecting tracks with a ghost probability of  
171 larger than  $x\%$  will retain  $x\%$  of all fake tracks.

172 In addition any update of the ghost probability training will have the same behaviour  
173 and thus the optimal working points of algorithms using the ghost probability algorithm  
174 will remain unchanged to a good approximation.

## 175 1.6 Category classifiers

176 Fake tracks produced by different pattern recognition algorithms might have different  
177 track properties. It might therefore be beneficial to train separate neural networks for  
178 the two long track reconstruction algorithms at LHCb. On simulated events, the physics  
179 performance of two separate networks does not differ from the physics performance of a  
180 single network. Similarly, different networks for different T station regions have been tested  
181 (one for tracks in the OT, IT, and the overlap region), without significant performance  
182 gain. The possible reasons are twofold. Firstly, the network already knows which T station  
183 tracker a track went through due to the hit counts in the individual subdetectors. Secondly,  
184 for the different algorithms, it is possible the common track fit is more indicative for  
185 whether a track is a fake or not, than the pattern recognition strategy.

186 Consequently, a single network for all pattern recognition algorithms in the entire  
187 detector is deployed.

## 188 1.7 Training sample

189 The LHCb track reconstruction needs to be able to handle a wide range of LHC running  
190 conditions. At the time of preparing for data taking in 2015 it was not clear whether the  
191 LHC would operate at 25 ns bunch spacing or 50 ns bunch spacing. Four scenarios were  
192 simulated:

- 193 • 25 ns bunch spacing,  $\nu = 1.6$
- 194 • 25 ns bunch spacing,  $\nu = 1.9$
- 195 • 50 ns bunch spacing,  $\nu = 1.6$
- 196 • 50 ns bunch spacing,  $\nu = 2.7$

197 where  $\nu$  is the average number of  $pp$  interactions per bunch crossing. Each event is required  
198 to contain at least one  $b\bar{b}$  pair.

199 The scenarios differ, for what concerns the track reconstruction, significantly in detector  
200 occupancy and spillover in the OT. This may lead to different behaviours of fake track  
201 reconstruction and require different network trainings for different running conditions. The  
202 necessity for having different network trainings is assessed by training networks for each  
203 of the running conditions, with all other training parameters fixed, and evaluating the  
204 networks on one of the samples and their discriminating powers are compared. Figure 4

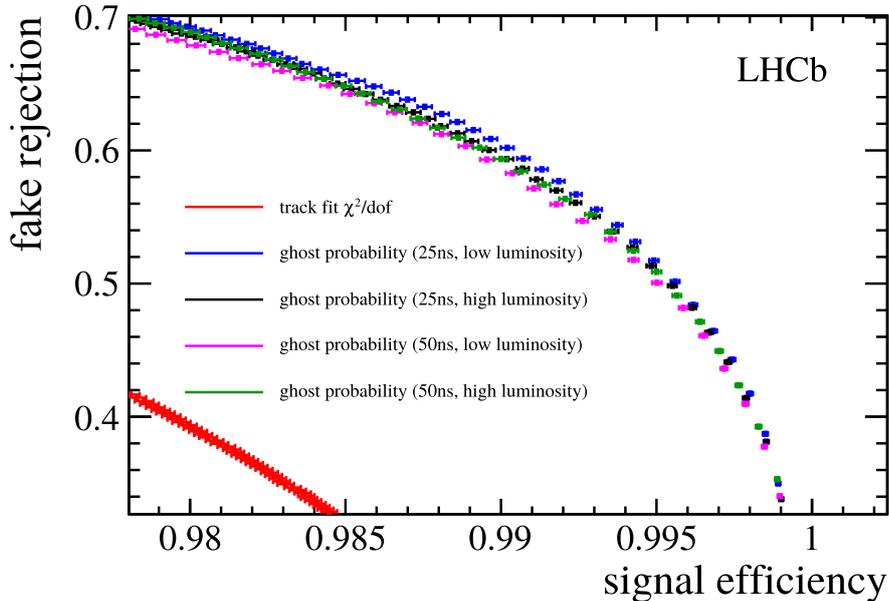


Figure 4: ROC curves for fake track discriminating neural networks (blue, black, magenta, and green), trained for different LHC running conditions, evaluated for 25 ns bunch spacing,  $\nu = 1.6$ . The red points are the ROC curve for the track fit reduced  $\chi^2$ , which performs significantly worse than any of the neural networks.

205 shows the receiver operating characteristics (ROC) curves for the 25 ns sample at low  
 206 pile-up. The discriminating powers of the four networks do not largely differ and thus for  
 207 simplicity only a single network (trained at the favoured scenario of 25 ns bunch spacing  
 208 at low pile-up) is deployed.

## 209 2 Physics Validation

210 The data taking strategy of LHCb in Run II involves the application of the same track  
 211 reconstruction in the software trigger as in the offline data processing. This goal can  
 212 only be achieved within the time constraints of the software trigger by applying the ghost  
 213 probability in the trigger. This ghost probability reduces the rate of fake tracks entering  
 214 the particle identification and combinatorics of decay reconstructions and thereby saves  
 215 more time than the computation of the ghost probability.

216 It must therefore be ensured that the full physics program of LHCb can be done with  
 217 tracks passing the ghost probability, and that there is no corner of phase space or particle  
 218 species, which is rejected by the ghost probability.

219 The computation of the track fit  $\chi^2$  was last revised in 2015 between data taking  
 220 at 50 ns and 25 ns bunch spacing, [16]. For conclusive validations, tracks in Run I data  
 221 and from 2015 data with 50 ns bunch spacing are refitted before computing the ghost  
 222 probability.

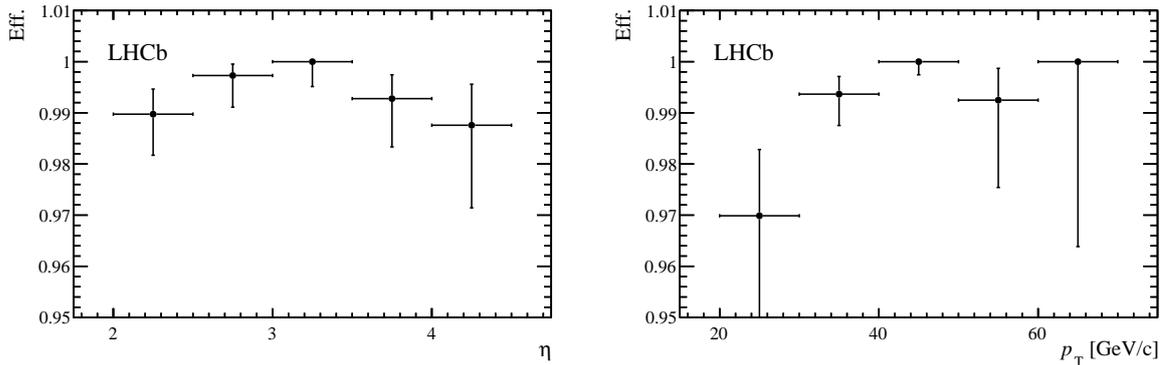


Figure 5: Efficiency for  $Z \rightarrow \mu\mu$  tracks to pass the ghost probability, in data from 2015 at a bunch spacing of 50 ns.

## 2.1 High momentum tracks

Due to their low cross section, stable charged particles at high momenta, in the momentum range of  $Z \rightarrow \mu\mu$  decays, are absent in the training data. This could lead to a low selection efficiency for very high momentum tracks.

In the early measurement period (corresponding to the first  $6 \text{ pb}^{-1}$ ) in 2015 at a bunch spacing of 50 ns, the nominal 2015 pattern recognition was used without application of the ghost probability. Refitting the candidate tracks from  $Z \rightarrow \mu\mu$  decays in that period allows to assess the performance of the ghost probability for very high momentum tracks. The measured efficiency in Fig. 5 shows that the absence of very high momentum tracks in the training data does not lead to a low efficiency.

## 2.2 Electron reconstruction

Electrons are more challenging to reconstruct than the standard candles ( $Z \rightarrow \mu\mu$  or  $D \rightarrow K\pi$ ). At the same time, it can be expected that the response of the ghost probability for electrons differs from that for other particles as electrons undergo more multiple scattering.

It is assumed that the reconstruction of converted photons as electron pair is the most vulnerable part for the following reasons. The photon conversion can happen “late” in the VELO leaving only few hits. In addition, the  $e^+e^-$  pair has a small opening angle which could lead to hit ambiguities in the VELO pattern reconstruction. It should be noted that the branching fraction of  $B_s^0 \rightarrow K^*\gamma$  is anyhow low and the statistical sensitivity of an analysis of this decay would immediately suffer from further efficiency loss.

The 50 ns early data of 2015 does not correspond to enough integrated luminosity to obtain a satisfying estimation of the consequences of a cut on the ghost probability on converted photons. For this reason, the tracks of  $B_s^0 \rightarrow K^*\gamma$  candidates from Run I – using a simplified version of the selection presented in Ref. [17] without Bremsstrahlungs correction – are refitted using the track fit configuration as used in 2015 and the ghost

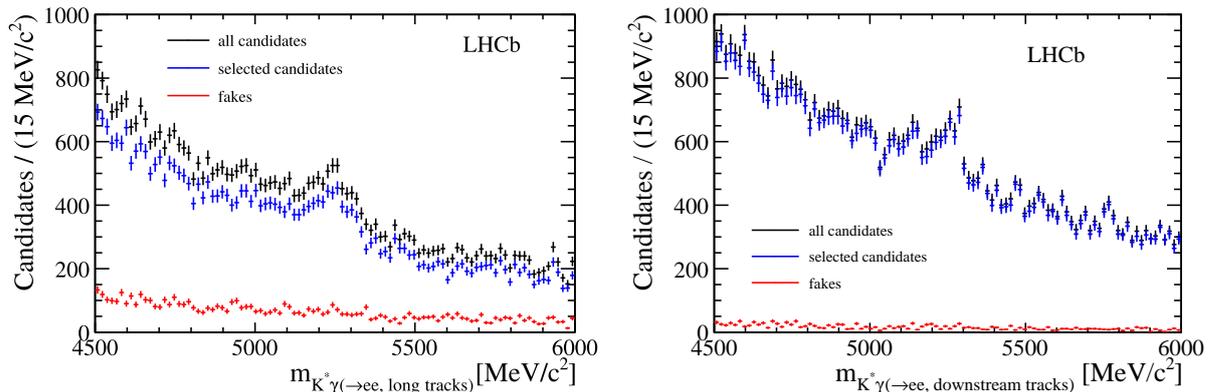


Figure 6:  $B_s^0 \rightarrow K^* \gamma$  candidates, where the photon is reconstructed as pair of electron tracks, using long tracks (left) and using downstream tracks (right). The candidates prior to a cut on the ghost probability of the electron tracks are shown in black, those passing the ghost probability in blue, and those candidates rejected by the ghost probability in red. No signal loss is visible in the rejected candidates.

249 probability is evaluated. The invariant mass spectrum shown in Fig. 6 shows candidates  
 250 without the application of a cut on the ghost probability, those passing, and those failing;  
 251 both for converted photons reconstructed as pair of downstream tracks and as pair of long  
 252 tracks. To the statistical precision of this test, no signal loss is visible.

### 253 2.3 Validation with 25 ns data

254 A cut on the ghost probability is included in the standard track reconstruction since data  
 255 taking at 25 ns bunch spacing started. To investigate the behaviour of the ghost probability  
 256 in real data with 25 ns bunch spacing, events are re-reconstructed without a cut on the  
 257 ghost probability. Under the assumption, that most  $K_s^0$  are part of the underlying event  
 258 and most triggered events containing  $K_s^0$  would have been triggered without those  $K_s^0$ ,  $K_s^0$   
 259 are used as probe of the ghost probability.

260 The invariant mass spectrum of  $K_s^0$  candidates after re-reconstruction is shown in  
 261 Fig. 7 for both long tracks and downstream tracks. In both cases,  $K_s^0$  candidates are  
 262 reconstructed from two opposite charged pions which are compatible with originating  
 263 from a common vertex, which satisfy fiducial momentum requirements, and which are  
 264 significantly displaced from any primary collision vertex. In either case, the background  
 265 contribution is largely reduced when rejecting events where at least one of the tracks has  
 266 a ghost probability of larger than 0.4. There is no signal visible in the events rejected. It  
 267 is concluded that no physics signal is lost due to the application of the ghost probability  
 268 within the statistical sensitivity of the test in the kinematic spectrum of the selected  $K_s^0$ .

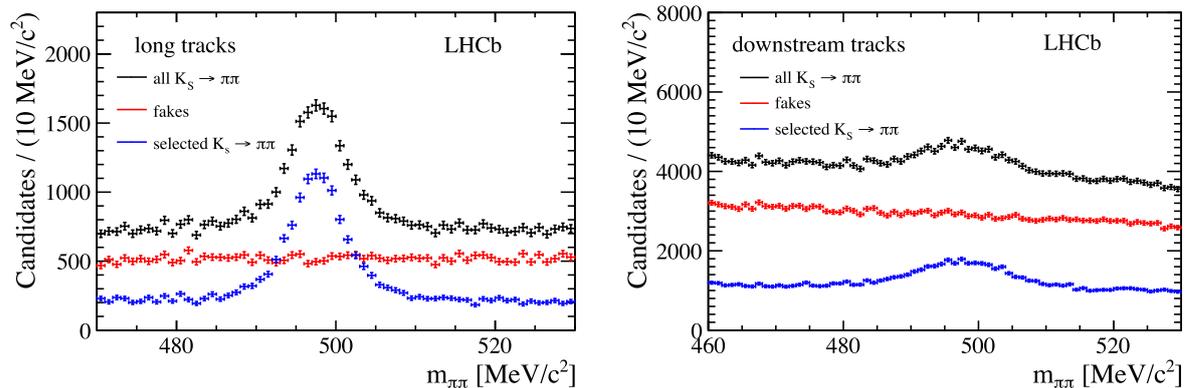


Figure 7:  $K_S^0 \rightarrow \pi\pi$  invariant mass spectrum for events reconstructed without using the ghost probability in the track selection; using long tracks (left) and downstream tracks (right). Candidates for which at least one track fails the default ghost probability requirement are shown in red and do not exhibit a signal contamination. The remaining candidates are shown in blue.

## 269 2.4 Decay time acceptance

270 The study of long lived particles ( $b$  and  $c$  hadrons) is the major part of the LHCb physics  
 271 program. It must therefore be ensured that the ghost probability does not reject particles  
 272 from displaced vertices at a higher probability than particles from primary collisions (which  
 273 have a higher prevalence in the training).

274 To evaluate a possible decay time bias of the ghost probability, for each reconstructed  
 275 particle in simulated events with a  $b\bar{b}$  production, the average of the true decay time of the  
 276 ancestor particles is determined. When rejecting tracks which fail the ghost probability  
 277 criterion, the average decay time of the parent particle changes by  $(1.5 \pm 2.0)$  fs. This  
 278 is smaller than the statistical sensitivity of this test, and smaller than the systematic  
 279 uncertainty to which the lifetime bias of the LHCb reconstruction is known [18].

## 280 3 Impact on trigger CPU consumption

281 The ghost probability was gradually introduced in the LHCb software trigger over Run II.  
 282 In the early measurement period in 2015 at 50 ns bunch spacing, it was evaluated in the  
 283 software trigger but not used to select tracks for trigger decisions. For the rest of 2015 all  
 284 tracks in the second stage (HLT2) had to pass a ghost probability selection. In 2016, an  
 285 additional requirement was introduced in the first stage of the software trigger (HLT1).

286 Applying the ghost probability in HLT1, reduces the rate of events which pass the  
 287 first stage, and therefore the HLT2 reconstruction needs to reconstruct 4% less events. In  
 288 combination with the ghost probability requirement in HLT2, the RICH PID is executed  
 289 for 22% fewer tracks and the CPU used for it is reduced by 23%. The number of  
 290 combinations, in the 2-body topological trigger, is reduced by 69%, resulting in 53% fewer  
 291 trigger candidates. The CPU consumption for combinatorics for all trigger lines is reduced

292 by 58 %.

293 The evaluation of the ghost probability itself “costs” 0.2 % of the HLT CPU budget  
294 (in roughly equal parts for HLT1 and HLT2), which underlines the overall benefit of its  
295 application.

296 For the entire software trigger this results in a reduced CPU consumption of 16 % –  
297 assuming that roughly half of the current HLT farm costs  $\mathcal{O}(2\text{ M CHF})$ , this is equivalent  
298 to 640 k CHF.

299 Lastly, if the ghost probability was removed from the HLT and the remaining setup  
300 not adapted, the output rate would increase by 36 %.

## 301 4 Outlook

302 The current networks are trained for the track reconstruction for data taking in 2015 at  
303 25 ns bunch spacing, using the latest simulations available at the time. Retraining is  
304 advisable for significant updates in the track reconstruction before the ghost probability  
305 (*i.e.* the pattern recognition and the track fit) and to account for expected changes in the  
306 detector performance due to radiation damage. Physics performance gains can also be  
307 expected with improved machine learning techniques or event simulations.

308 Additional separation between “good” tracks and fake tracks could be gained by  
309 testing more detailed information about the hit expectations in active layers. The old  
310 ghost probability used only the number of expected hits per subdetector, instead of their  
311 locations.

312 The current training is purely based on simulated events, the domain adaptation  
313 approach from Ref. [19] is not applied as it currently does not lead to an improved fake  
314 track rejection. The ghost probability network is retrained using good tracks and fake  
315 tracks from simulated events and unlabelled tracks from real events with the Caffe software  
316 framework [20]<sup>4</sup>. In addition to the network with domain adaptation, a conventional  
317 network is trained to disentangle effects from the training algorithm (replacing TMVA by  
318 Caffe) and network architecture (adding a gradient reversal layer and domain classifier).  
319 This working point of the network responses is chosen to retain the same number of  
320 candidates in a  $K_S^0 \rightarrow \pi\pi$  selection as the application of the nominal ghost probability.  
321 From the invariant mass distribution in Fig. 8, it can be seen that the TMVA network  
322 and the two networks trained in Caffe yield close to identical physics performance; the  
323 data points of the network with domain adaptation are almost entirely covered by the  
324 data points of the Caffe network without domain adaptation. This does not rule out that  
325 domain adaptation can improve the physics performance of the ghost probability in the  
326 future.

---

<sup>4</sup>An implementation of the method suggested in Ref. [19] has been provided by its authors in the Caffe framework.

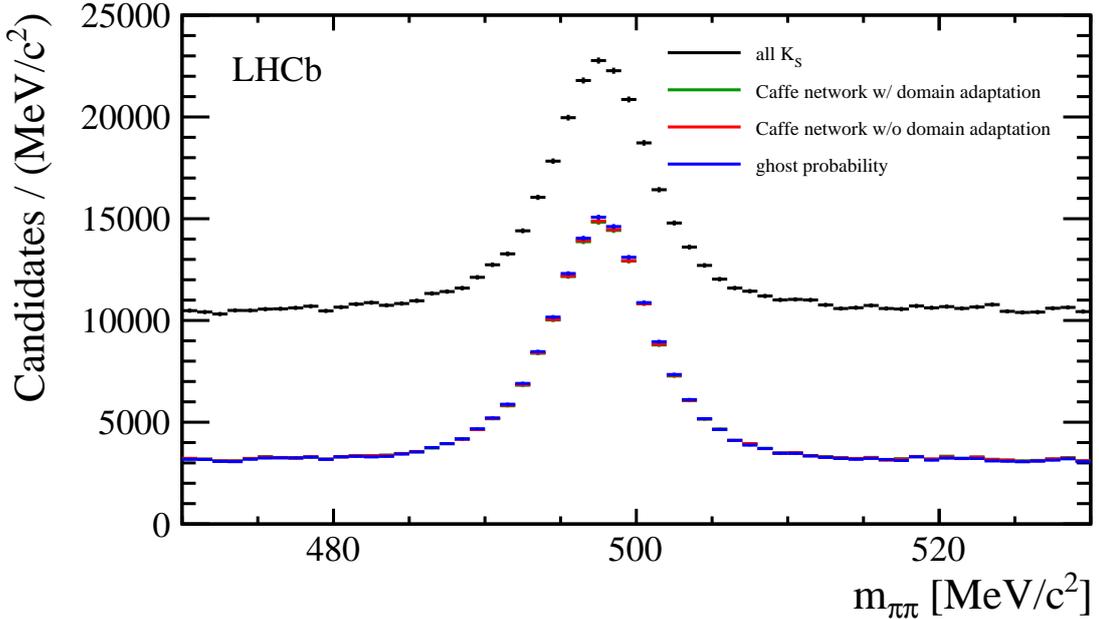


Figure 8: Comparison of the  $K_S^0$  selection with the nominal ghost probability (blue), a network trained with the Caffe package (red), and a network trained with domain adaptation (green, covered by red). The working points of the Caffe networks are chosen to retain the same numbers of candidates. Hardly any performance difference is visible.

Table 1: Callgrind benchmark comparisons of different activation functions. Fields with n/a have not been evaluated or are not available with AVX intrinsics. The activation function used by the ghost probability is marked with (\*).

function	default compiler options	AVX vectorisation by hand
$\tanh$	19,355,124,355	n/a
$\frac{1}{1+e^{-x}}$	21,140,125,632	n/a
$\frac{x}{\sqrt{1+x^2}}$ (*)	415,121,741	195,121,939
$\frac{x}{1+ x }$	395,121,798	195,104,759
$\max(0, x)$	155,095,875	115,095,891

327 The current network relies on auto-vectorisation. The methods suggested by Ref. [21]  
328 lead to tenfold improvement of the CPU cycle count of the neural network implementation  
329 [14] using AVX [22] intrinsic commands. This approach has not been followed up to ensure  
330 platform independence of the ghost probability.

331 The current activation function in the neural network is  $\frac{x}{\sqrt{1+x^2}}$ . The rectified linear  
332 unit  $\max(0, x)$  or  $\frac{x}{1+|x|}$  are expected to be even faster, as listed in Table 1.

## 5 Conclusion

The ghost probability is introduced as a the default method of reducing the number of fake tracks in the LHCb reconstruction. Its CPU performance improvement by a factor 10 helped its deployment for offline and online reconstruction in Run II of the LHC. The reduction of fake tracks in the particle identification and combinatorics of decay reconstruction greatly reduces the demand of computing resources of the software trigger and enables LHCb to use identical reconstructions online and offline. Validations at different phase space points reveal no adverse side effects of applying the ghost probability centrally.

## Acknowledgements

We express our gratitude to our colleagues in the LHCb collaboration who provided suggestions and helped in the implementation, especially Angelo Di Canto, Helge Voss, Manuel Schiller, Gerhard Raven, Chris Jones; from the Electronic Vision(s) group (Kirchhoff-Institute for Physics, Ruprecht-Karls-Universität Heidelberg, Heidelberg, Germany) Eric Müller; and from the LHC Physics and New Particles group (Institute for Theoretical Physics, Ruprecht-Karls-Universität Heidelberg, Heidelberg, Germany) Johann Brehmer.

## References

- [1] LHCb collaboration, R. Aaij *et al.*, *Measurement of the track reconstruction efficiency at LHCb*, JINST **10** (2015) P02007, [arXiv:1408.1251](#).
- [2] LHCb collaboration, R. Aaij *et al.*, *LHCb detector performance*, Int. J. Mod. Phys. **A30** (2015) 1530022, [arXiv:1412.6352](#).
- [3] S. Broghi *et al.*, *REAL TIME CALIBRATION public document*, , in preparation.
- [4] J. Weidendorfer, M. Kowarschik, and C. Trinitis, *A Tool Suite for Simulation Based Analysis of Memory Access Behavior*, in *Computational Science - ICCS 2004, 4th International Conference, Kraków, Poland, June 6-9, 2004, Proceedings, Part III* (M. Bubak, G. D. van Albada, P. M. A. Sloot, and J. Dongarra, eds.), vol. 3038 of *Lecture Notes in Computer Science*, pp. 440–447, Springer, 2004. <http://www.valgrind.org/docs/pubs.html>.
- [5] R. Aaij *et al.*, *Performance of the LHCb Vertex Locator*, JINST **9** (2014) P09007, [arXiv:1405.7808](#).
- [6] R. Arink *et al.*, *Performance of the LHCb Outer Tracker*, JINST **9** (2014) P01002, [arXiv:1311.3893](#).

- 365 [7] J. Brehmer, J. Albrecht, and P. Seyfert, *Ghost probability: an efficient tool to remove*  
366 *background tracks*, <https://cds.cern.ch/record/1478372>. LHCb internal note  
367 LHCb-INT-2012-025.
- 368 [8] LHCb collaboration, R. Aaij *et al.*, *Observation of  $B_c^+ \rightarrow J/\psi D_s^+$  and  $B_c^+ \rightarrow J/\psi D_s^{*+}$*   
369 *decays*, Phys. Rev. **D87** (2013) 112012, [arXiv:1304.4530](https://arxiv.org/abs/1304.4530).
- 370 [9] A. Hoecker *et al.*, *TMVA: Toolkit for Multivariate Data Analysis*, PoS **ACAT** (2007)  
371 040, [arXiv:physics/0703039](https://arxiv.org/abs/physics/0703039).
- 372 [10] M. Schiller, H. Voss, and L. Moneta, *code available on Jira ticket: fast tanh imple-*  
373 *mentation*, <https://sft.its.cern.ch/jira/browse/ROOT-7054>.
- 374 [11] Wikipedia. Sigmoid function, 2015.
- 375 [12] P. Seyfert and H. Voss, *code available on Jira ticket: TActivation... implementations*,  
376 <https://sft.its.cern.ch/jira/browse/ROOT-7062>.
- 377 [13] X. Glorot and Y. Bengio, *Understanding the difficulty of training deep feedforward*  
378 *neural networks*, in *Proceedings of the Thirteenth International Conference on Ar-*  
379 *tificial Intelligence and Statistics* (Y. W. Teh and M. Titterton, eds.), vol. 9 of  
380 *Proceedings of Machine Learning Research*, (Chia Laguna Resort, Sardinia, Italy),  
381 pp. 249–256, PMLR, 13–15 May, 2010.
- 382 [14] P. Seyfert, *code available on github project TMVA-MLP*, [https://github.com/](https://github.com/pseyfert/tmva-mlp)  
383 [pseyfert/tmva-mlp](https://github.com/pseyfert/tmva-mlp).
- 384 [15] C. M. Bishop, *Pattern recognition and machine learning*, Information science and  
385 statistics, Springer, New York [u.a.], 10. (corrected at 8th printing) ed., 2009; J.-H.  
386 Zhong *et al.*, *A program for the Bayesian Neural Network in the ROOT framework*,  
387 Comput. Phys. Commun. **182** (2011) 2655, [arXiv:1103.2854](https://arxiv.org/abs/1103.2854).
- 388 [16] M. Heß, *Multiple scattering in track reconstruction*, [https://cds.cern.ch/record/](https://cds.cern.ch/record/1957764)  
389 [1957764](https://cds.cern.ch/record/1957764). LHCb internal note LHCb-INT-2014-043.
- 390 [17] L. Beaucourt, E. Tournfier, M. N. Minard, and J. F. Marchand,  *$B^0 \rightarrow K^* \gamma(e^+e^-)$  and*  
391  *$B_s^0 \rightarrow \phi \gamma(e^+e^-)$  analysis status*, in *2nd Radiative decays @LHCb Workshop* (A. Oyan-  
392 guren Campos *et al.*, eds.), 2015. <https://indico.cern.ch/event/375424/>.
- 393 [18] LHCb collaboration, R. Aaij *et al.*, *Measurements of the  $B^+$ ,  $B^0$ ,  $B_s^0$  meson and  $\Lambda_b^0$*   
394 *baryon lifetimes*, JHEP **04** (2014) 114, [arXiv:1402.2554](https://arxiv.org/abs/1402.2554).
- 395 [19] Y. Ganin and V. Lempitsky, *Unsupervised Domain Adaptation by Backpropagation*,  
396 ArXiv e-prints (2014) [arXiv:1409.7495](https://arxiv.org/abs/1409.7495).
- 397 [20] Y. Jia *et al.*, *Caffe: Convolutional Architecture for Fast Feature Embedding*,  
398 [arXiv:1408.5093](https://arxiv.org/abs/1408.5093).

- 399 [21] V. Vanhoucke, A. Senior, and M. Z. Mao, *Improving the speed of neural networks on*  
400 *CPUs*, <https://research.google.com/pubs/archive/37631.pdf>.
- 401 [22] N. Firasta *et al.*, *Intel AVX: New frontiers in performance improvements and*  
402 *energy efficiency*, Intel white paper **19** (2008) 20, [https://software.intel.](https://software.intel.com/sites/default/files/m/9/e/1/8/1/16820-Intel_AVX_New_Frontiers_in_Performance_Improvements_and_Energy_Efficiency_WP.pdf)  
403 [com/sites/default/files/m/9/e/1/8/1/16820-Intel\\_AVX\\_New\\_Frontiers\\_in\\_](https://software.intel.com/sites/default/files/m/9/e/1/8/1/16820-Intel_AVX_New_Frontiers_in_Performance_Improvements_and_Energy_Efficiency_WP.pdf)  
404 [Performance\\_Improvements\\_and\\_Energy\\_Efficiency\\_WP.pdf](https://software.intel.com/sites/default/files/m/9/e/1/8/1/16820-Intel_AVX_New_Frontiers_in_Performance_Improvements_and_Energy_Efficiency_WP.pdf).